

Conceptual Dependency로부터 한글 SEMANTIC NETWORK의 생성

문 병로, 김 진형

한국과학기술원, 전산학과

요약

본 논문은 중간표현으로써 R C Schank의 Conceptual Dependency(CD)를 사용하여 이를 입력으로 하여 한글의 Semantic Network을 생성하는 시스템에 대하여 기술한다. 본 시스템은 CD를 입력으로 받아 Discrimination Network에 의해 동사론 선택한 다음 이 동사론의 case frame을 채워 Semantic Network을 생성한다. 본 논문에서는 위 시스템의 설계를 위해 한글 동사에 대한 Discrimination Network과 case frame의 slot filling 방법 등이 제시된다.

I. 서론

인간의 모든 의사소통 활동에 있어 그 매체가 되는 기초나 언어가 가능한 한 제약이 없는 자연스런 것이길 바라는 것은 효율성과 편리성을 추구하는 인간의 경향에 매우 자연스럽게 부합된다. 이는 computer science에서 다른 어떤 분야보다도 많이 요구하는 것이기도 하다. 고급 프로그래밍 언어의 parsing 기술이 상당한 수준에 이르자 인간이 일상생활에서 사용하는 자연 언어를 parsing하고자 하는 당연한 움직임이 일어났다.

자연 언어 처리의 역사는 크게 문법적인 접근 방식과 의미적인 접근 방식으로 나눌 수 있다. 문법적인 접근 방식은 N Chomsky로부터 태동했다고 보는 것이 일반적인 견해다. Chomsky는 1957년 그의 유명한 저서 'Syntactic Structure'에서 변형·생성 문법을 세상에 소개하여 체계적인 문법적 parsing의 동기를 제공하였다. 그러나 자유분방하고 과격적인 요소가 많은 인간의 언어 활동을 제약이 많은 문법적 틀에 맞추는 것은 당연히 많은 어려움과 무리가 따랐다. 이에 자연 언어 처리에 대한 회의론이 얼마간 학계를 지배하게 되나 1970년에 W Wood의 ATN grammar가 발표되면서 다시 활기를 띠기 시작했다[11]. 1975년에는

A Colmerauer에 의해 DCG(Definite Clause Grammar)가 제안되는데 이는 문법적 parsing에 logic의 개념을 첨가시킨 것이다[5]. 문법적 parsing방법이 상당한 수준에 이르렀지만 그 자체로 만족할 만한 자연 언어 처리를 한다는 것은 본질적으로 많은 미흡함을 안고 있었다.

의미적인 접근 방식은 1968년에 C.Fillmore의 유명한 논문 'Case for cases'로부터 시작된다. Fillmore는 어순을 중요시하는 문법적 접근 방식의 한계를 인식, 문장의 중심을 동사로 보고 각 동사 고유의 case frame을 설정, 입력 문장으로부터 slot filling함에 의해 parsing하는 방식을 제안했다[6]. Fillmore의 case grammar가 문장의 표면 구조와 전혀 무관하지는 못한 데 비해 R.Schank는 문장의 표면 구조와 완전한 독립을 이룬 case grammar인 CD (Conceptual Dependency)를 제안했다[7]. 이것은 인간의 사고 작용에 대한 정성스럽고 공들인 통찰의 결과로만 나올 수 있는 성질의 것이다. 본 논문은, 자연 언어 문장을 완전히 이해했다고 말하려면 이를 parsing한 결과를 문장의 표층 구조와 전혀 무관한 의미만의 표현인 CD와 같은 것으로 나타내야 한다는 확신으로부터 출발하고 있다.

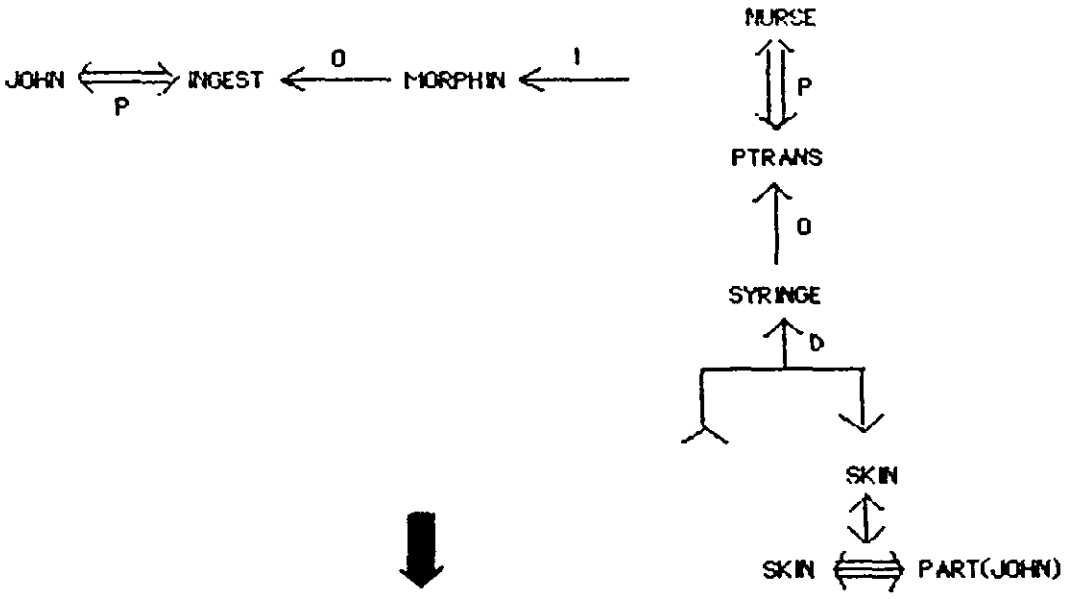
본 시스템은 CD를 입력으로 받아 Discrimination Network에 의해 동사를 선택한 다음 이 동사 고유의 case frame을 채워 Semantic Network을 만들어 내는 것이다. 본 시스템에서는 한글 동사의 선택을 위한 Discrimination Network과 한글 동사에 대한 case frame 및 slot filling 방법 등이 제시된다.

II. Hangul Half Generator의 구조와 기능

CD를 심층 표현으로 하는 시험적인 한글 생성기(Hangul Generator)의 flow는 다음과 같다.

- 1 CD로부터 동사를 선택한다.
2. 이 동사 고유의 case frame에 slot filling을 한다
이 결과로 Semantic Network이 만들어진다.
- 3 이 Semantic Network을 parsing하여 최종적인 한글 문장을 만들어 낸다

영어의 경우 N.Goldman이 이와 유사한 방법을 적용하여 성공한 바 있다[4]. 본 시스템은 위 중 1과 2를 대상으로 하고 있다 이를 위해서는 CD를 LISP notation으로 표현하는 방



```
(ACT INGEST
  ACTOR (CAT HUMAN CALLED JOHN)
  OBJECT (CAT PHYS-OBJ CALLED MORPHIN)
  TIME PAST
  INSTRMT (ACT PTRANS
    OBJECT (CAT PHYS-OBJ CALLED SYRINGE)
    TO (CAT PHYS-OBJ WHOSE JOHN CALLED SKIN)
    ACTOR (CAT HUMAN OCCUPATION NURSE)
  )
)
```

그림1

법, 한글 동사 선택에 필요한 Discrimination Network, 한글 동사군에 대한 case frame 및 각 case frame에 대한 slot filling 방법 등이 개발되었다. 위 1, 2를 구현한 시스템을 Hangul Half Generator라 명명했다

1 Conceptual Dependency와 이의 LISP상에서의 표현법

CD는 개념(문장의 의미)을 나타내는 데 있어 표층문장이 속한 언어적 구조에 전혀 무관한 표현방식이다 즉 문장 자체를 그대로 옮기는 것이 아니라 그 문장을 서술하는 사람의 머리속에서 그 문장을 만들어 내는 실제적인 구조를 흉내내본 것이다 CD의 기본 골격은 act나 state중의 하나인 것이 보통이다. act는 행위가 기본이 되는 개념(예로 ...먹다 ...보다 등)이고 state는 상태가 기본이 되는 개념(예로 ...이다 ...파랗다 등)이다. 대부분의 개념은

act를 기본으로 한다. act에는 11개의 primitive action (PTRANS, MTRANS, INGEST, ATTEND, ...)이 있다[7].

그림1은 graphic notation으로 표현된 CD를 LISP상에서의 표현법으로 나타낸 예이다.

2. 한글 동사의 Discrimination Network

Discrimination Network(이하 D-Net라 칭함)은 binary tree로서 그림2에서 보는 것처럼 internal node들은 predicate들로 되어 있고 terminal node들은 선택되는 동사를 가리키도록 되어 있다. 입력 CD의 기본 골격으로부터 여러 개의 D-Net들 중 어느 것을 검색할 것인지 결정한 다음 선택된 D-Net의 root node로부터 internal node들을 따라 일련의 테스트를 실시하여 terminal node에 도달한다. 임의의 internal node의 predicate를 수행하여 참이면 right child로 거짓이면 left child로 이동한다.

시스템에서 대상으로 하는 동사는 의미상의 중복을 최대한 억제한 minimal cover로 정함을 원칙으로 한다. 이 minimal cover속에서 '파랴다'와 '푸르다'처럼 미묘한 어감의 차이같은 것의 식별은 기대할 수 없다.

D-Net내의 predicate들은 예외없이 입력 CD의 어느 한 구성요소의 성질을 테스트하는 것이다. 이를 위해서는 테스트하려고 하는 CD내의 부분을 찾아낼 수 있는 방법이 정형화되어야 한다. 이를 field-specification이라고 하고 {ACTOR, OBJECT, MOBJECT, TO, FROM, TIME, MODE, }의 원소들의 list로 구성된다. field-specification의 값은 다음의 algorithm에 의해 추출된다.

입력 : field-specification과 대상 conceptualization

출력 : 입력된 field-specification이 가리키는 conceptualization

algorithm :

1. VALUE <- 입력된 conceptualization.
FS <- 입력된 field-specification.
2. VALUE <- (car FS)가 가리키는 field의 conceptualization.
3. FS <- (cdr FS)

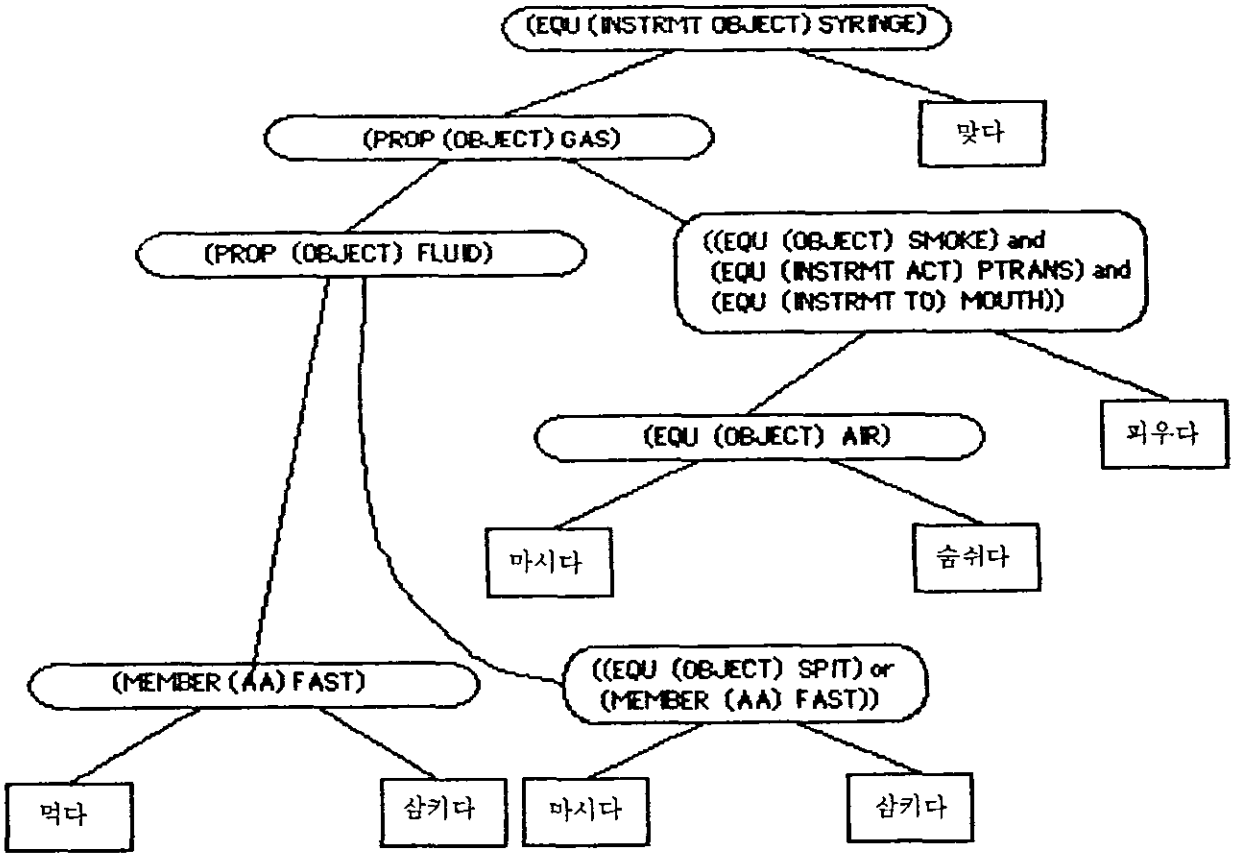


그림2

4. if (null FS)
 then return current VALUE
 else goto step 2.

한글의 INGEST D-Net를 그림2에 소개한다. predicate는 '(predicate-name field-specification comparand)'의 형태를 취하는데 comparand는 field-specification이거나 string이다.

3. 한글 동사에 대한 case frame과 slot filling방법

문장은 명제(proposition)와 서법(modality)으로 이루어진다.

'영이가 책을 보았다'

라는 문장이 있을 때 '영이가 책을 보' 부분이 명제이고 '았다' 부분이 서법에 해당한다.

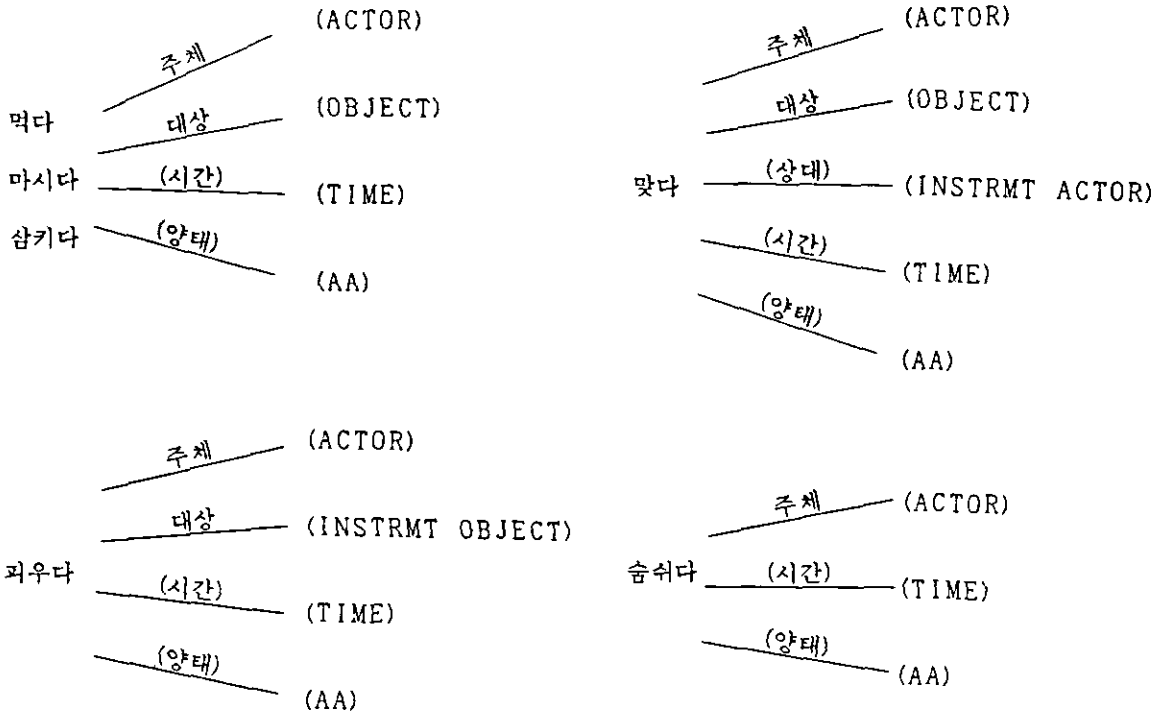


그림3

서법 '있다' 대신 '느냐'로 대체하면 의문문이 된다.

동사에 대해 제공되는 case frame은 문장이 아닌 명제 부분을 대상으로 하고 Semantic Network은 문장 전체를 대상으로 한다. 위의 예에서는 '보다'에 대해 '(주체 대상)'이라는 case frame이 제공된다. 여기에 위의 명제 부분을 맞추어 넣으면 '주체 = 영이, 대상 = 책'과 같이 slot filling이 된다 또한 여기에 '시제 = 과거'라는 것을 합치면 위 문장에 대한 Semantic Network이 만들어진다. 실제로 구현할 때는 명제 부분에 대한 case frame을 필수적으로 제공한 다음 서법에 대한 case frame을 필요한 만큼 제공한다.

그림3은 한글의 INGEST 동사들의 각 군들에 대한 case frame과 그 slot filling 방법을 나타낸 것이다 case frame의 설정 기준은 명제의 심층적 의미보다는 표층 구조과의 관련성에 중점을 두었다 이것은 본 시스템의 출력인 Semantic Network이 문장의 의미를 나타내는

역할보다는 표층 문장으로 parsing하기 위한 중간 단계의 표현 방법으로서의 역할이 더 중요한 것이란 사실을 생각하면 쉽게 납득할 수 있을 것이다. 한글에 대해 여러가지의 격 설정 방법이 제시되었으나 본 시스템에서는 기본적으로 김 일웅씨의 것을 골격으로 했다[1]. 그림 3의 왼편은 같은 형태의 case frame을 취하는 동사군이고 edge의 label은 격의 이름을 나타내며 각 edge의 오른편은 해당 격을 채우는 방법을 나타낸다. 오른편의 표현은 field-specification이다.

III. 결론

중간표현으로써 R.Schank의 Conceptual Dependency(CD)를 사용하여 이를 입력으로 한글의 Semantic Network을 만드는 시스템을 설명하였다. Semantic Network으로부터 한글 표층 문장으로서의 생성은 본 논문에서는 다루지 않았으나 ATN parsing을 이용하면 될 것이다. 영어의 경우는 R.Simmons가 ATN parsing을 조금 변형시켜 적용시켜 성공하였다[9] Semantic Network을 parsing하여 표층 문장을 만드는 시스템과 본 시스템을 결합하면 Conceptual Dependency로부터의 한글 생성 시스템이 만들어진다 이 시스템은 한글 의역 시스템의 생성부에 사용될 수 있고, 한글 분석 시스템의 error debugging용으로도 사용될 수 있으며 기존의 영어 분석기와 합치면 영·한 기계 번역 시스템, 일어 분석기와 합치면 일·한 기계 번역 시스템이 될 수 있다.

IV. 참고 문헌

1. 김 일웅. (1984). 풀이말의 결합가와 격, 한글 재186호.
2. 배 해수. (1977). 격의 연구, 어문론집 18.
3. Bruce, B. (1975). Case systems for natural language. *Artificial Intelligence* 6, 327 - 360.
4. Goldman, N. (1975). Conceptual Generation. In 'Conceptual Information Processing', R. schank. North Holland, Amsterdam.
5. Pereira, F. and Warren, D. (1980). Definite clause grammars for language analysis. *Artificial Intelligent* 13, 231 - 278.
6. Samlowski, w. (1976) Case grammar.

In 'Computational semantics', E. Charniak and V. Wilks(eds.),
North Holland, Amsterdam.

7. Schank, R. (1975). 'Conceptual Information Processing', North Holland, Amsterdam.
8. Schank, R. and Riesbeck, C. (1981). 'Inside computer understanding',
- 9 Simmons, R. and Slocum, J. (1972). Generating English discourse from semantic networks. CACM, Vol. 15, No. 10.
10. Simmons, R., (1973). Semantic networks : Their computation and use for understanding English sentences. In 'Computational Models of Thought and Language', R.Schank and K.Colby(eds.), Freeman, San Fransisco, Calif.
11. Woods, W., (1970). Transition network grammars for natural language analysis. CACM, Vol. 13, No. 10.